# RAG Fundamentals and Large-Scale Deployment

Dalmo Cirne
Workday, Inc.
dalmo.cirne@workday.com
https://dalmocirne.com

Dalmo Cirne is a professional with three decades of experience computer science, mathematics, and leadership & management. He has a has a degree in mathematics from SUNY (State University of New York) and is passionate about building products and enabling the next generation of leaders and individual contributors.

Throughout this career, Dalmo has seen the world changing at a fast pace and realized that just teaching what is already known is an insufficient condition for success. Knowledge itself has to evolve, adapt to new realities, and sometimes influence what things will become.

Among his achievements, Dalmo includes the concept, design, and implementation of one the first known automated cash handling and processing systems together with DeLaRue; his contributions to RiskMetrics and the participation in its IPO; the creation of one of the first mobile fertility applications (attributed to having helped with the first UK's iPhone baby); building Disney's presence in the smartphone space, starting ScoreCenter (now ESPN app)–the product was a success, earning its own TV commercial and winning the prestigious Mobile Marketing Award in Innovation.

Transitioning to startups in the early 2010s, Dalmo played critical roles in companies like mParticle, where he helped taking the company from zero to one and to catapult the CDP (Customer Data Platform) market. There, he built not only products, but teams as well. Dalmo began his journey of expanding his learnings about management and leadership, in addition to his technical knowledge. At Clarifai, he assembled and led the team that brought intelligent computer vision to the DoD (Department of Defense). His work there is classified and required him to obtain Secret level security clearance. In addition, with the success of the many leadership and management techniques he was using at Clarifai, that period came with an epiphany and became the milestone when the concept of his book, Management Streams, was born.

Dalmo joined Workday in 2020, at the beginning of the COVID-19 pandemic. There, he assembled the ML (Machine Learning) and AI (Artificial Intelligence) teams for financials. This was an especially important time, because the concepts of Management Streams were put to the test of running teams in an unprecedented way. Not only have they been successful growing the org from 3 to 30+ people, but also in building many products (e.g., Journal Insights, Expense Protect) released to customers, including several on the Fortune 500 list.

Dalmo is a recognized inventor with one patent granted, and three more filed and under evaluation.

## How do trustors build trust with trustees?

- This question comes from the paper “Fostering Trust and Quantifying Value of AI and ML,” which I co-authored and will be presenting later this week at this conference

## How do trustors build trust with trustees?

## Let us ask ChatGPT

- LLMs will try to make sense of questions you ask and will do their best to provide an answer—whether it is good or not

# How do trustors build trust with trustees?

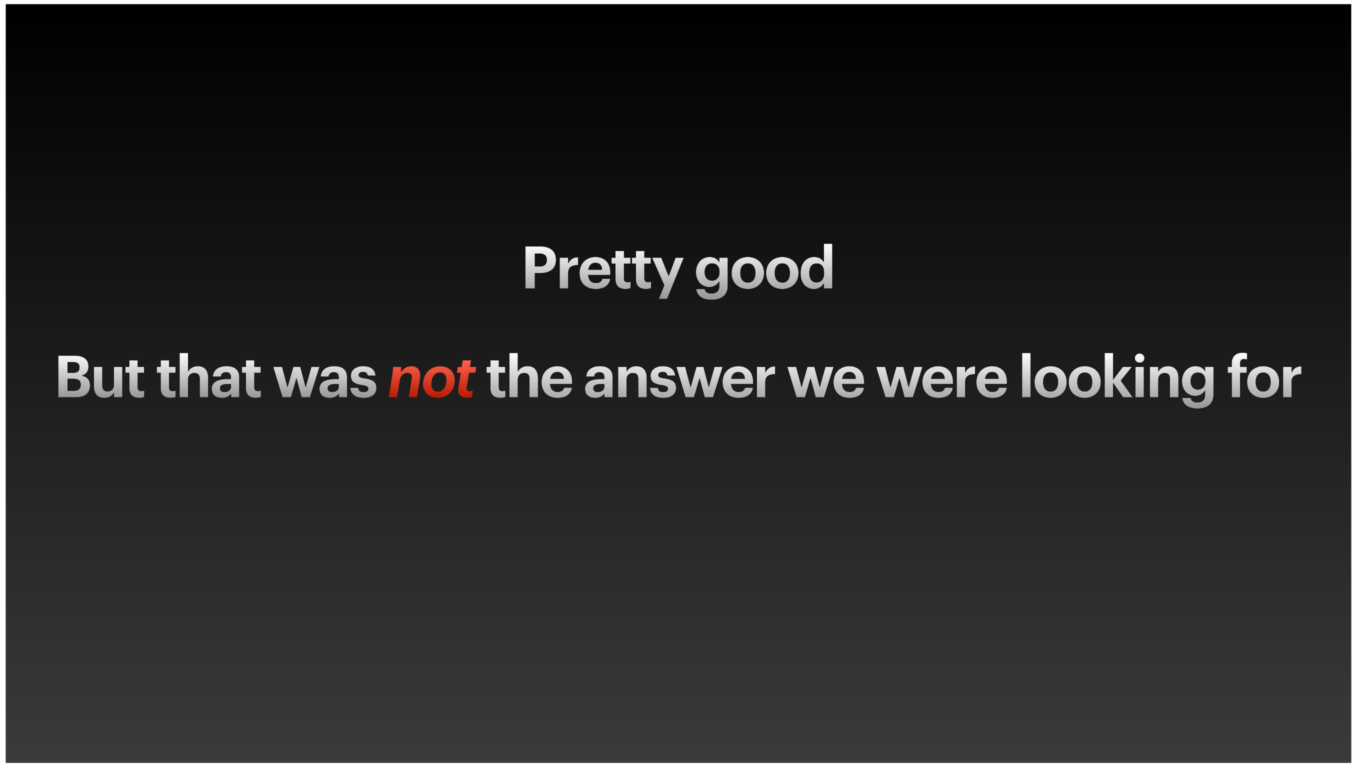
ChatGPT Answer: Trustors can build trust with trustees by:

1. Communicating openly and honestly: Trustors should communicate their expectations, concerns, and feedback openly and honestly with trustees. This helps to establish transparency and build a foundation of trust.

2. Setting clear expectations: Trustors should clearly outline their expectations for the trustee's role, responsibilities, and performance. This helps to ensure that both parties are on the same page and working towards the same goals.

3. Demonstrating reliability and consistency: Trustors can build trust by consistently following through on their commitments, being reliable, and demonstrating consistency in their actions and decisions.

4. Showing respect and appreciation: Trustors should show respect for the trustee's expertise, opinions, and contributions. Expressing appreciation for the




- The answer was sensible, well-constructed, and quite convincing
- Except that it was not the answer we were looking for

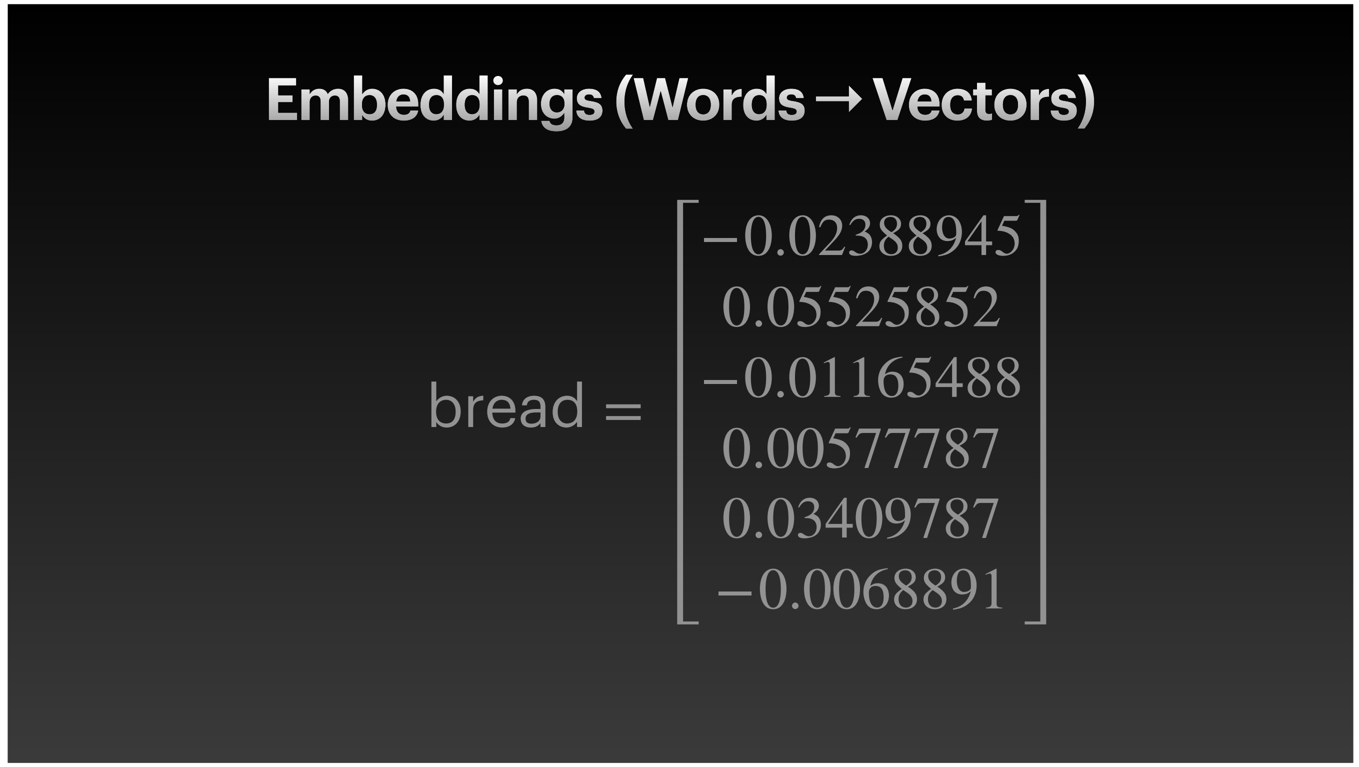
## How do we get better results?

We need to take a look at RAG and Semantic Chunking

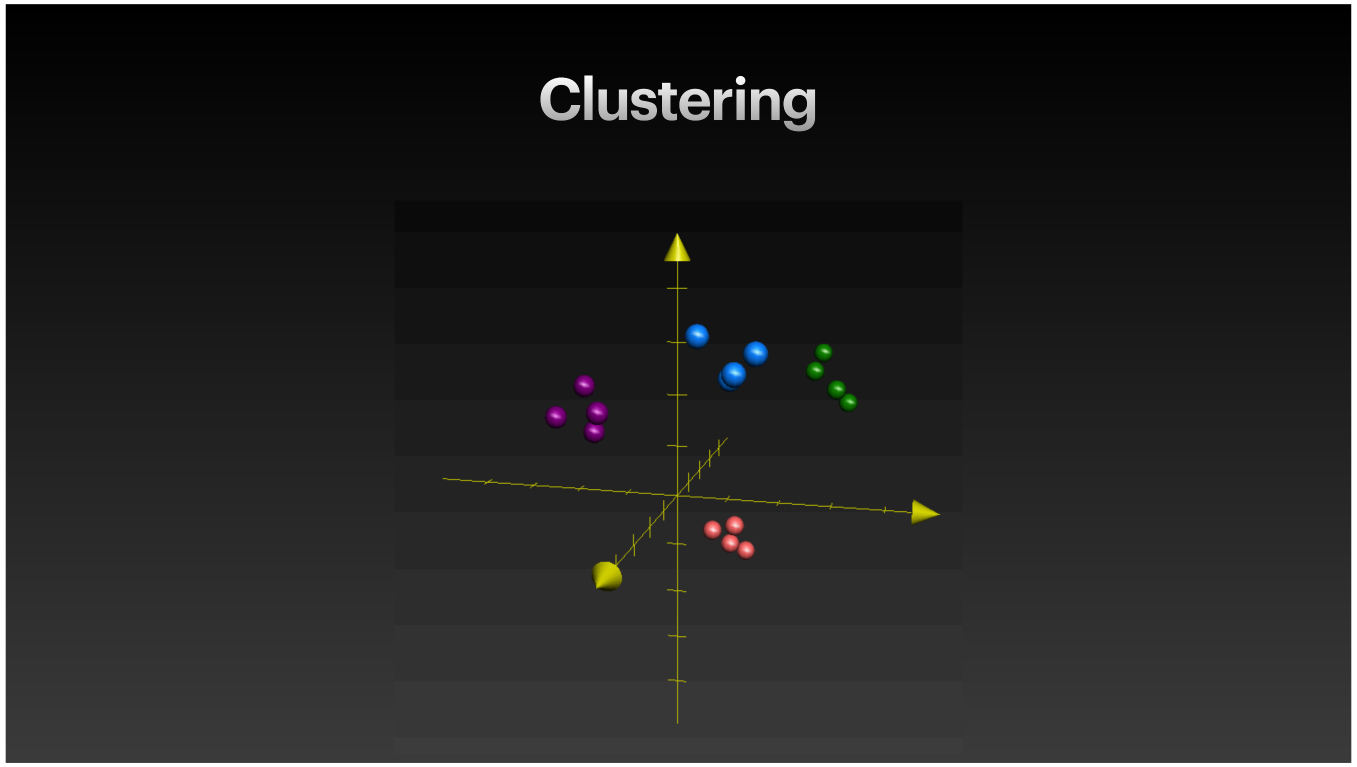
# Word Pairs

| Common | Uncommon |
|---|---|
| bread and butter | ephemeral tungsten |
| cause and effect | superstitious aardvark |
| customer service | supply and cons |
| sink or swim | yammering introvert |

- There word pairs that often appear together in text or common speech. It feels familiar when they are read or heard together
- On the other hand, when words are combined in ways that are not often seen together, it becomes immediately apparent that they are anomalous
- Claude Shannon noticed that and took it into consideration when he proposed what became to be known as Information Theory
    - Shannon, C. E. (1948). "*A mathematical theory of communication*". Bell System Technical Journal. 27 (3): 379–423, 623–656. doi:10.1002/j.1538-7305.1948.tb01338.x

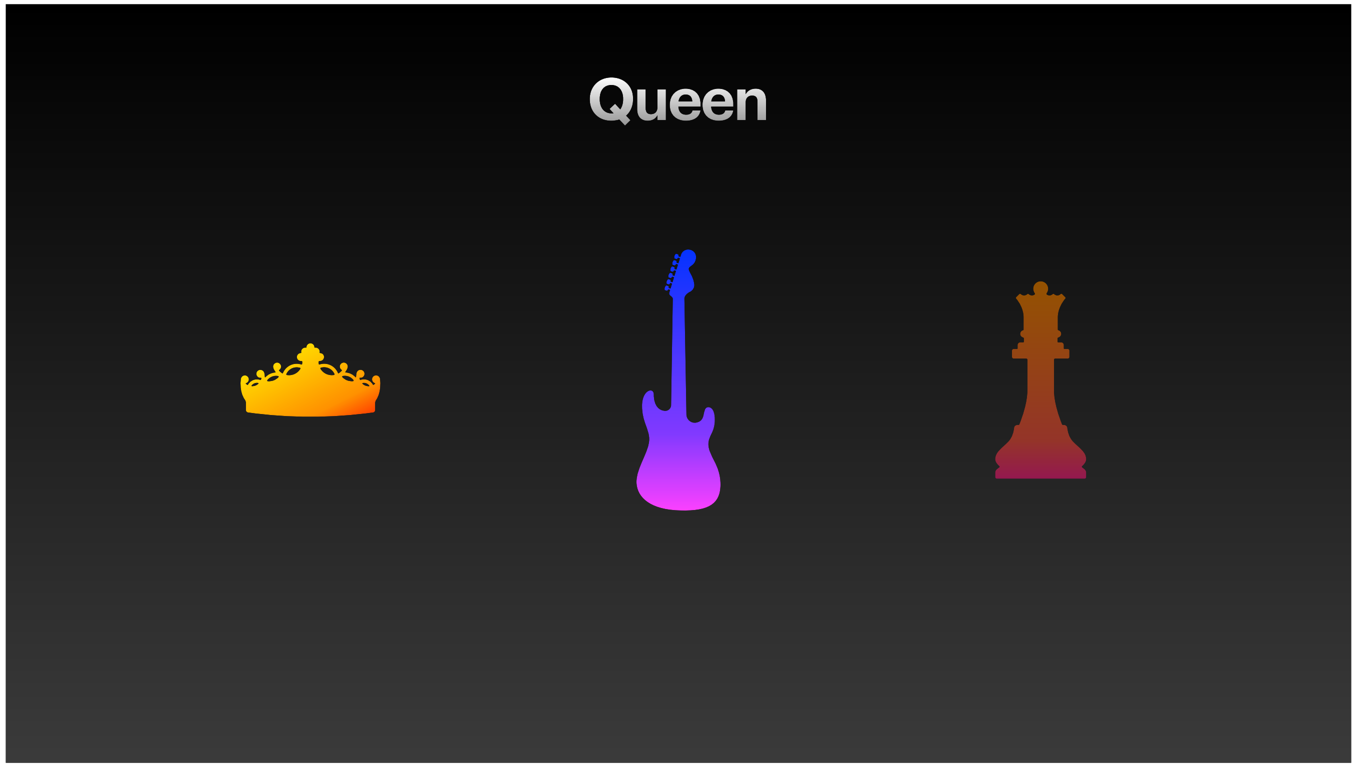



- It is necessary to represent media numerically, so it can be handled by computers
- Embeddings are numerical representations of an asset (e.g., text, image, sound)
- They are represented, stored, and used as vectors
- For simplicity, we are looking at a vector with just a few dimensions, but in reality those vectors have thousands of dimensions
- Natural Language Processing (NLP) tokenize characters, words, sentences, and subwords to create a vocabulary

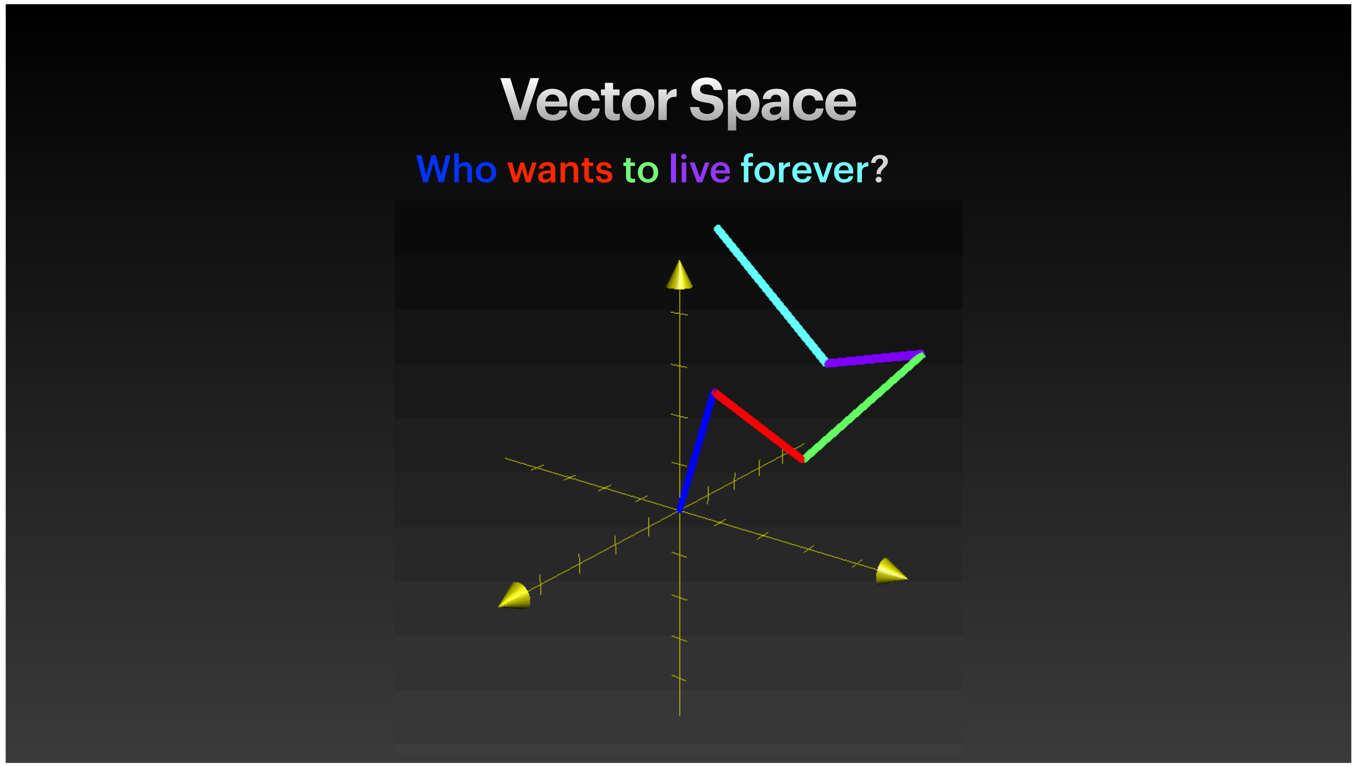



- As a model is being trained (tuning its weights), embeddings tend to settle in a way where clusterings form with semantic meaning
- An intuitive way of thinking about dot products, in this context, is that it measures how well vectors align


- What comes to mind when you read the word “Queen”?




- Do you think of a monarch, the rock band, or a chess piece?
- The meaning of words are informed by their surroundings




- The embeddings are not merely representing individual words. They encode information about the position of that word and carry context about the content

# RAG (Intuition)

- **Retrieval**: Fetch semantically relevant text from a database
- **Augmented**: Modify the prompt with the retrieved data
- **Generation**: Generate LLM output

The idea is to dynamically fetch the most relevant data, then inject it into to the LLM prompt for In-Context Learning (ICL)

# In-Context Learning

“GPT first produces meta-gradients according to the demonstration examples, and then these meta-gradients are applied to the original GPT to build an ICL model.”

“Experimental results show that in-context learning behaves similarly to explicit fine-tuning from multiple perspectives."

Source: https://doi.org/10.48550/arXiv.2212.10559

# In-Context Learning

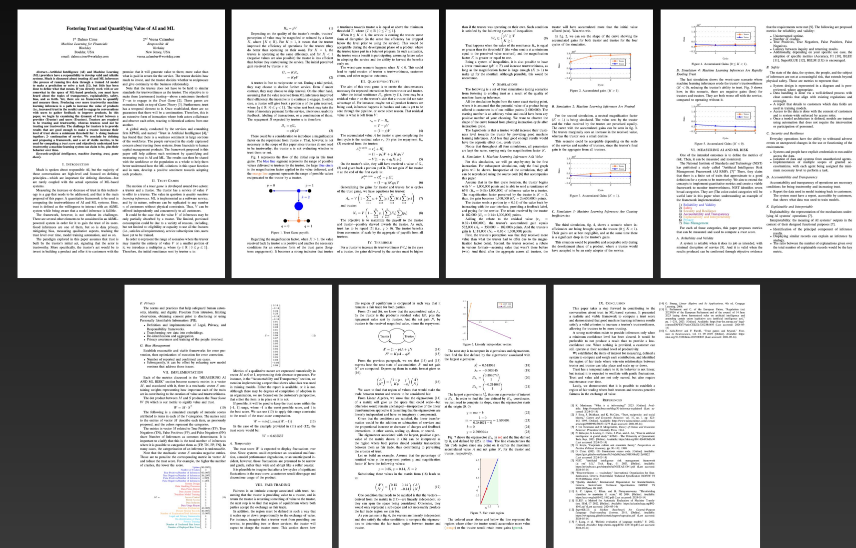
- Custom content can be provided to an LLM directly in the prompt
- There should be instructions indicating what is being asked, the content to be considered, and how the answer should be returned
- Consider the size of the context window
- Too much context could degrade results

```
prompt = f"""Answer the QUESTION based on the CONTEXT given.
If you do not know the answer and cannot find the answer in CONTEXT, say "I don't know."

QUESTION:
{user_question}

CONTEXT:
{rag_context}

ANSWER:
"""
```

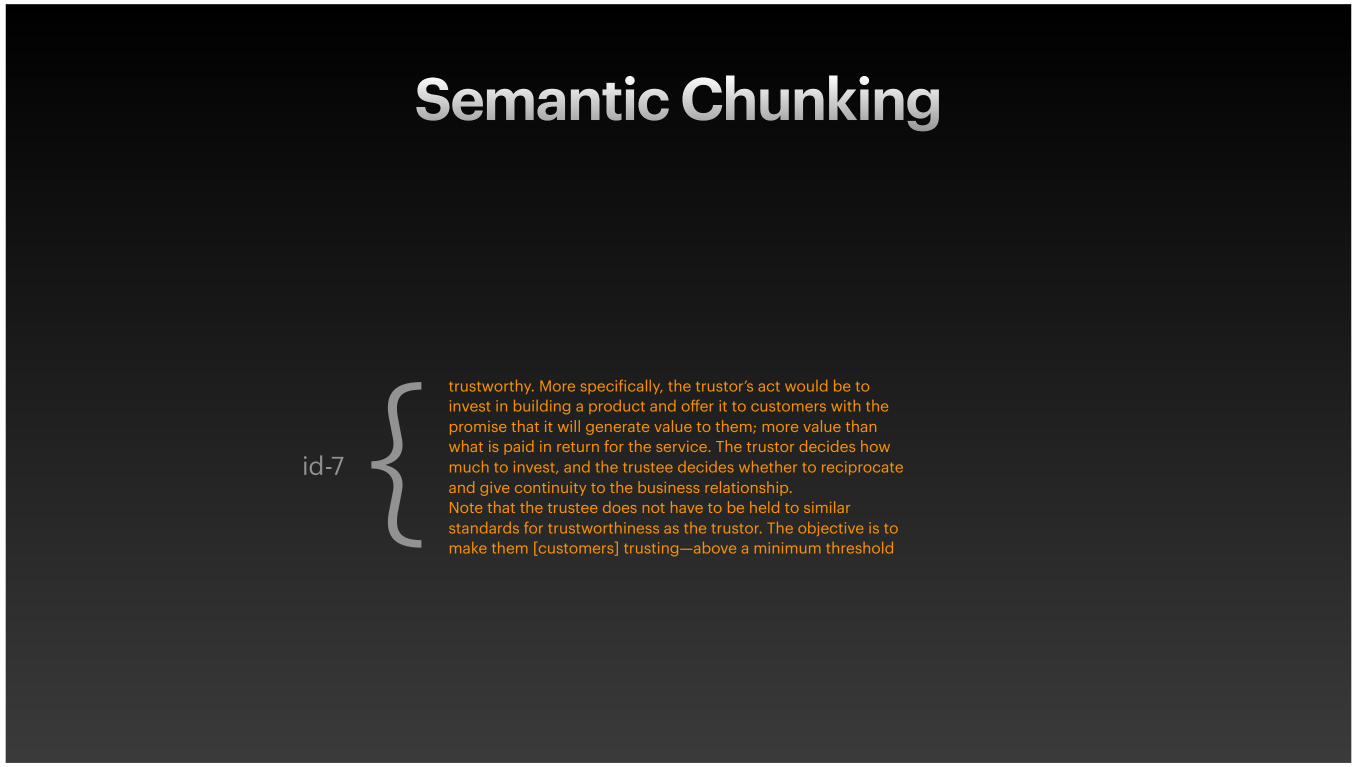
- Context window is the amount of tokens a model can receive as input. Its capacity influences how much information can be leveraged to run inferences.


- Vector databases are used to perform semantic similarity searches using techniques like the Approximate Nearest Neighbor (ANN) algorithm.

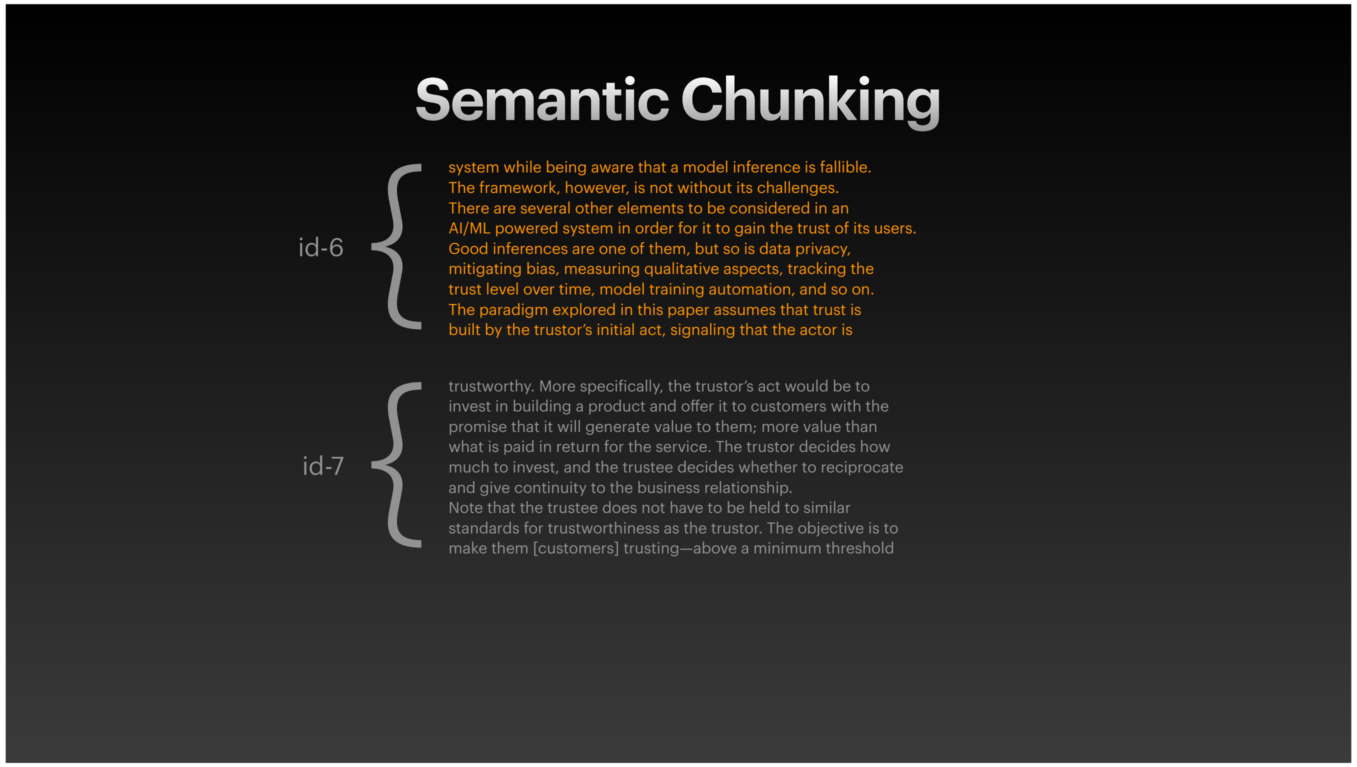
# Whole Document Chunking



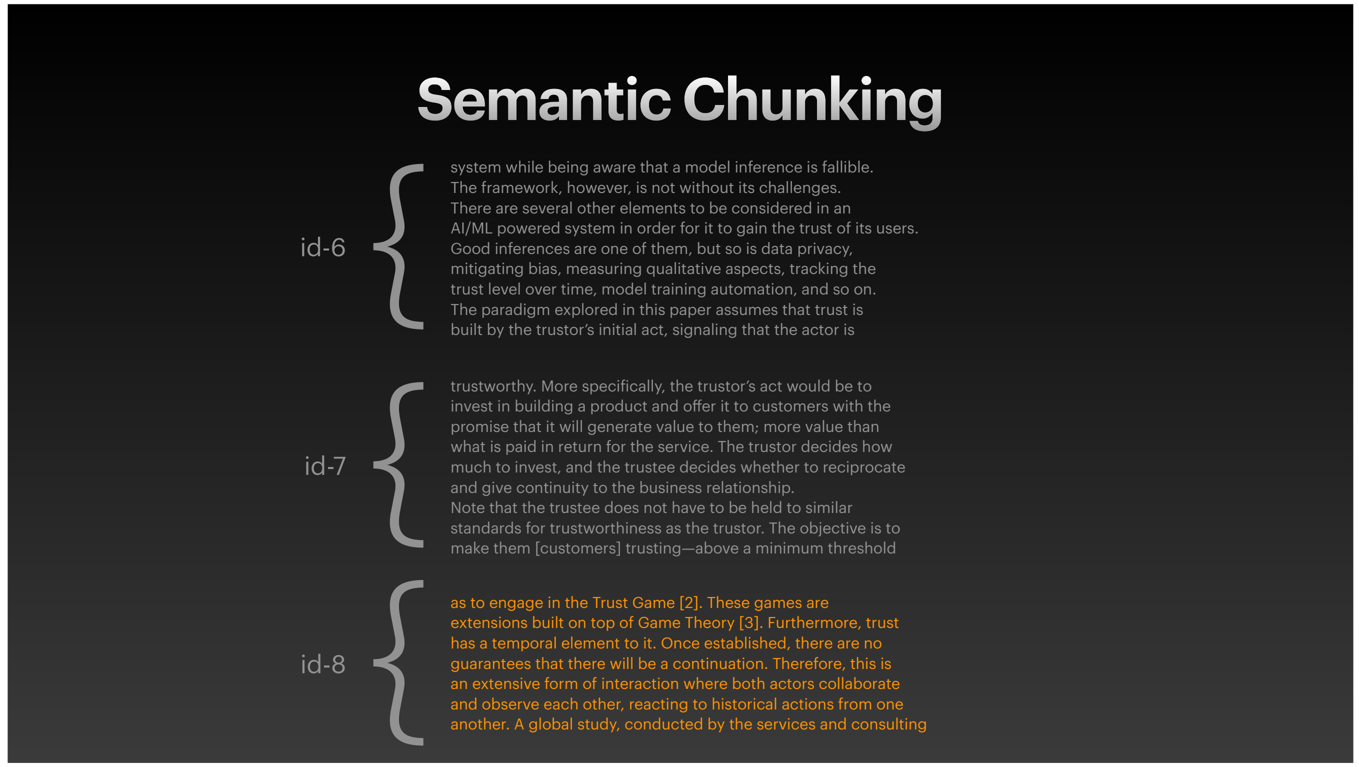
- We can give the LLM more content to be considered when running an inference
- One option is to provide the LLM the whole document as content… however, that may be too much (this screenshot is of the paper I mentioned earlier. You all should come to watch my talk tomorrow)
- Imagine a book about how to assemble a computer. It contains sections about CPUs, mother boards, displays, and so on. Now suppose you have a question on how to install a hard drive. Would you read the section about keyboards or would you go straight to the hard drives one?



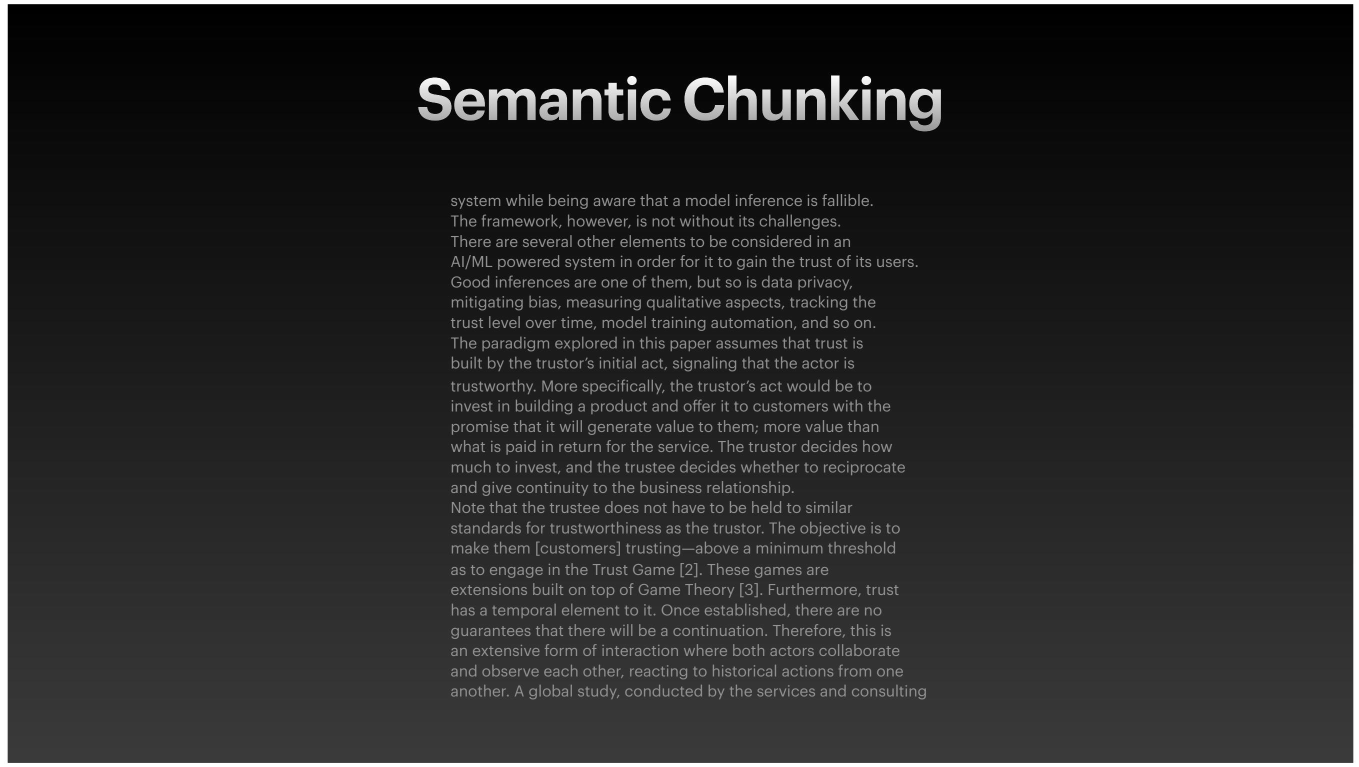

- Another option, is to limit the content to the portions that would be most relevant to the LLM
- The “chunks” of content would be semantically relevant to what is being asked
- Each chunk is assigned a unique identifier. id-7 is the chunk (and respective text) associated with the embeddings that is closest, in the vector space, to the query
- This is a naive implementation with the sole purpose of explaining the concept. It is not suitable for production environments. There are several tools out there that do a great job optimizing chunk sizes



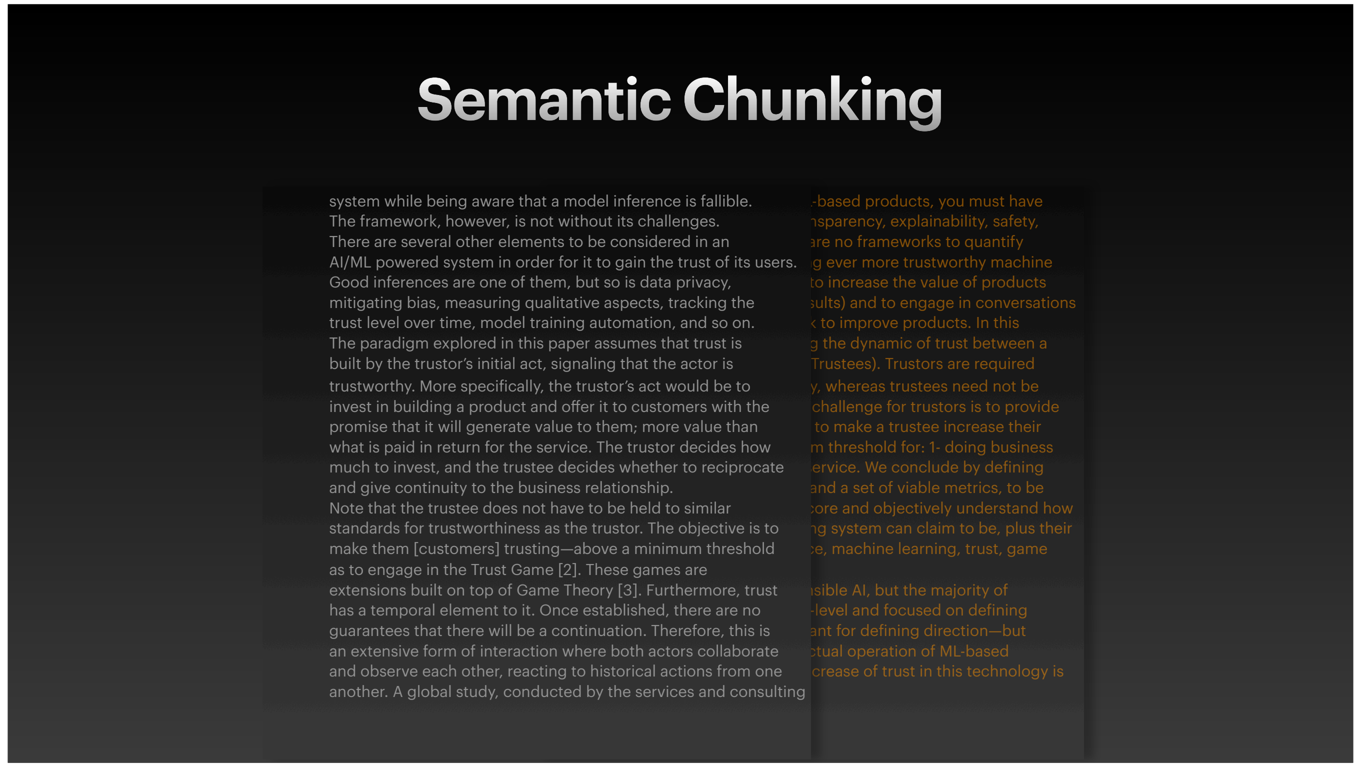

- Although id-7 is the closest chunk to the query, it may not contain enough information
- A good technique is to fetch the chunks immediate before and after, because they may have complimentary and import information
- There is a good change that the information surrounding a chunk is also relevant




- Following the same principle, id-8 is the chunk immediately after id-7




- We concatenate chunks id-6, id-7, and id-8 and use this text to populate the prompt to the LLM
- Another point worth mentioning is that because chunks' unique identifiers are being fetched from the vector database, it is possible to cite sources used in inferences




- In addition to just the chunk closest to the query, we can retrieve the top-n chunks and their immediate neighbors
- How many chunk top-n? That is a bit of an empirical art. Often fetching the top 2 or 3 are enough, but your mileage may vary

# Prompt

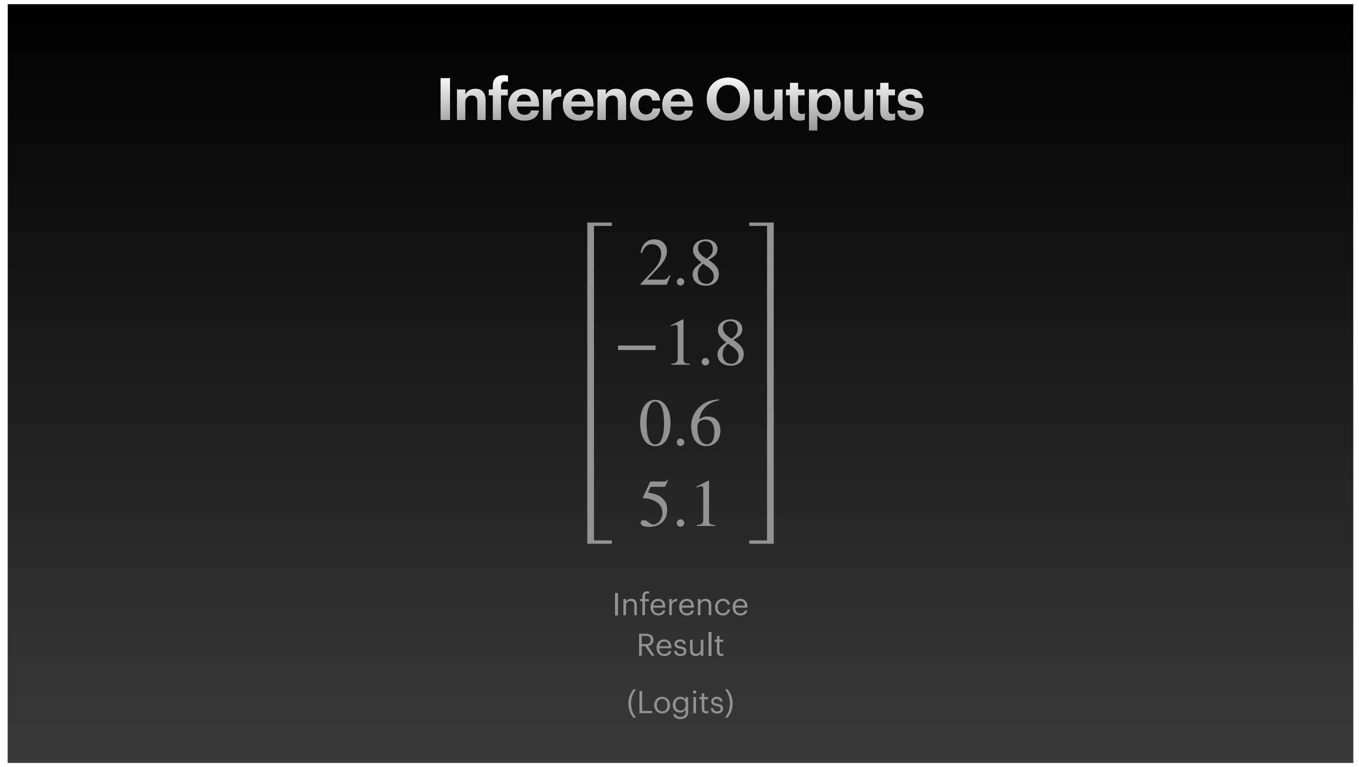
"""Answer the QUESTION based on the CONTEXT given.
If you do not know the answer and cannot find the answer in CONTEXT, say "I don't know."

QUESTION:
How do trustors build trust with trustees?

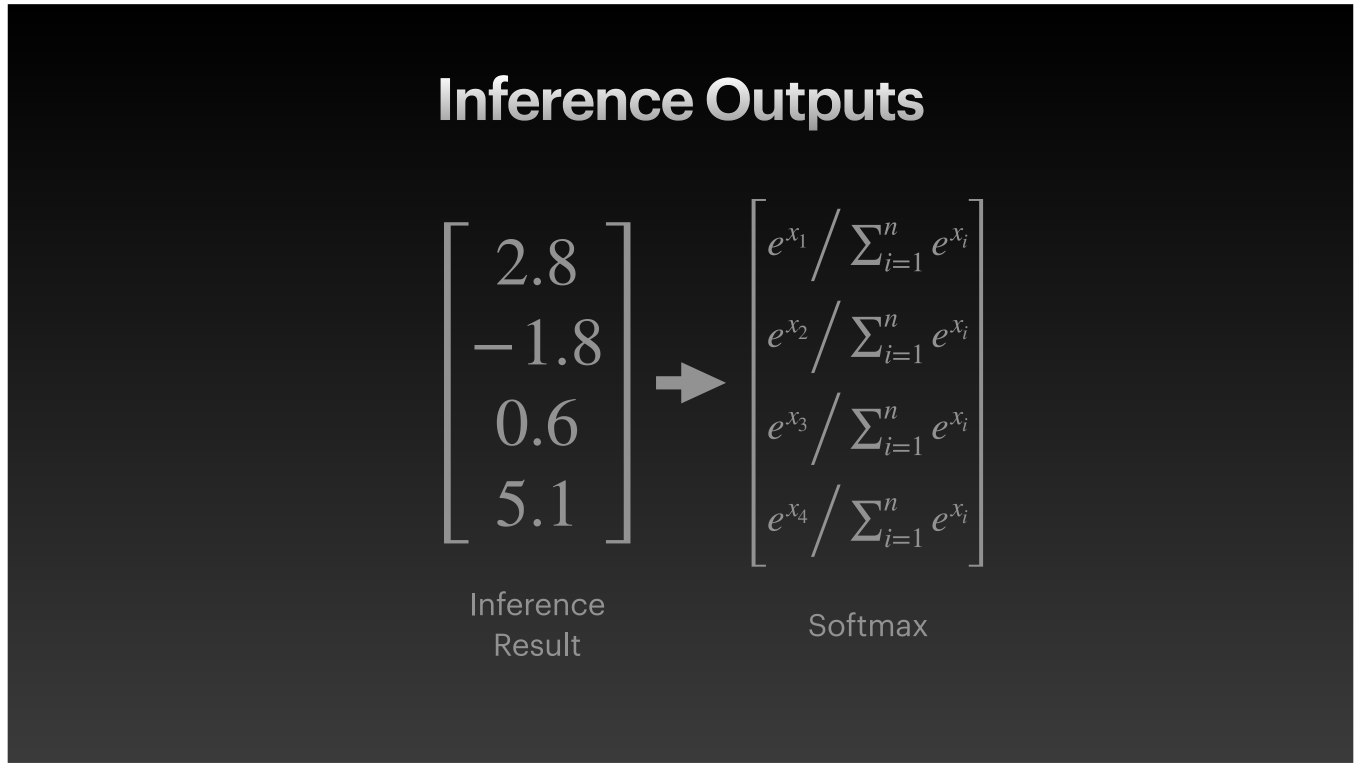
CONTEXT:
system while being aware that a model inference is fallible. The framework, however, is not without its challenges. There are several other elements to be considered in an AI/ML powered system in order for it to gain the trust of its users. Good inferences are one of them, but so is data privacy, mitigating bias, measuring qualitative aspects, tracking the trust level over time, model training automation, and so on. The paradigm explored in this paper assumes that trust is built by the trustor's initial act, signaling that the actor is trustworthy. More specifically, the trustor's act would be to invest in building a product and offer it to customers with the promise that it will generate value to them; more value than what is paid in return for the service. The trustor decides how much to invest, and the trustee decides whether to reciprocate and give continuity to the business relationship. Note that the trustee does not have to be held to similar standards for trustworthiness as the trustor. The objective is to make them [customers] trusting—above a minimum threshold as to engage in the Trust Game [2]. These games are extensions built on top of Game Theory [3]. Furthermore, trust has a temporal element to it. Once established, there are no guarantees that there will be a continuation. Therefore, this is an extensive form of interaction where both actors collaborate and observe each other, reacting to historical actions from one another. A global study, conducted by the services and consulting

ANSWER:
"""

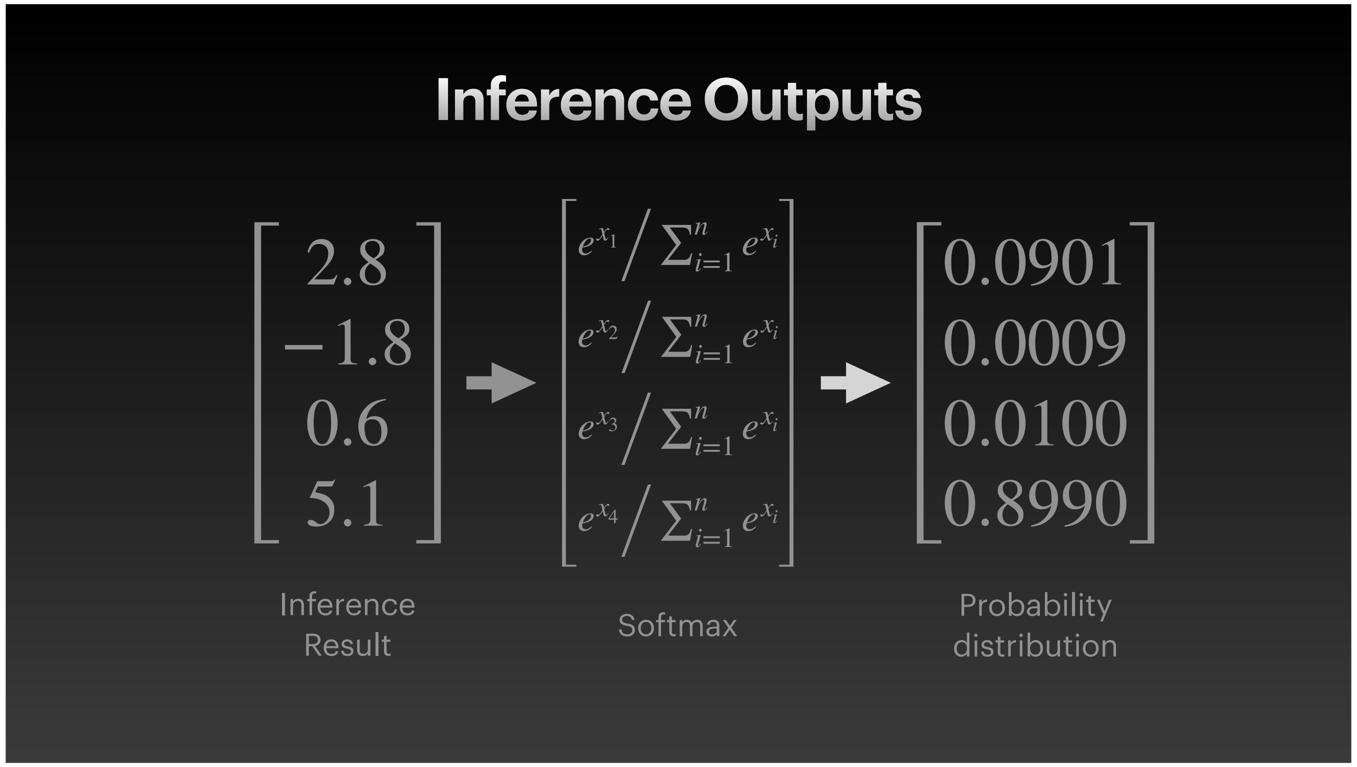
- The prompt in itself is another important element to pay attention to
- It has to be constructed in a way that instructs the LLM about intent, includes the original question, augments it with the retrieved content, and specifies expectations about the answer
- It is worth experimenting with different formats



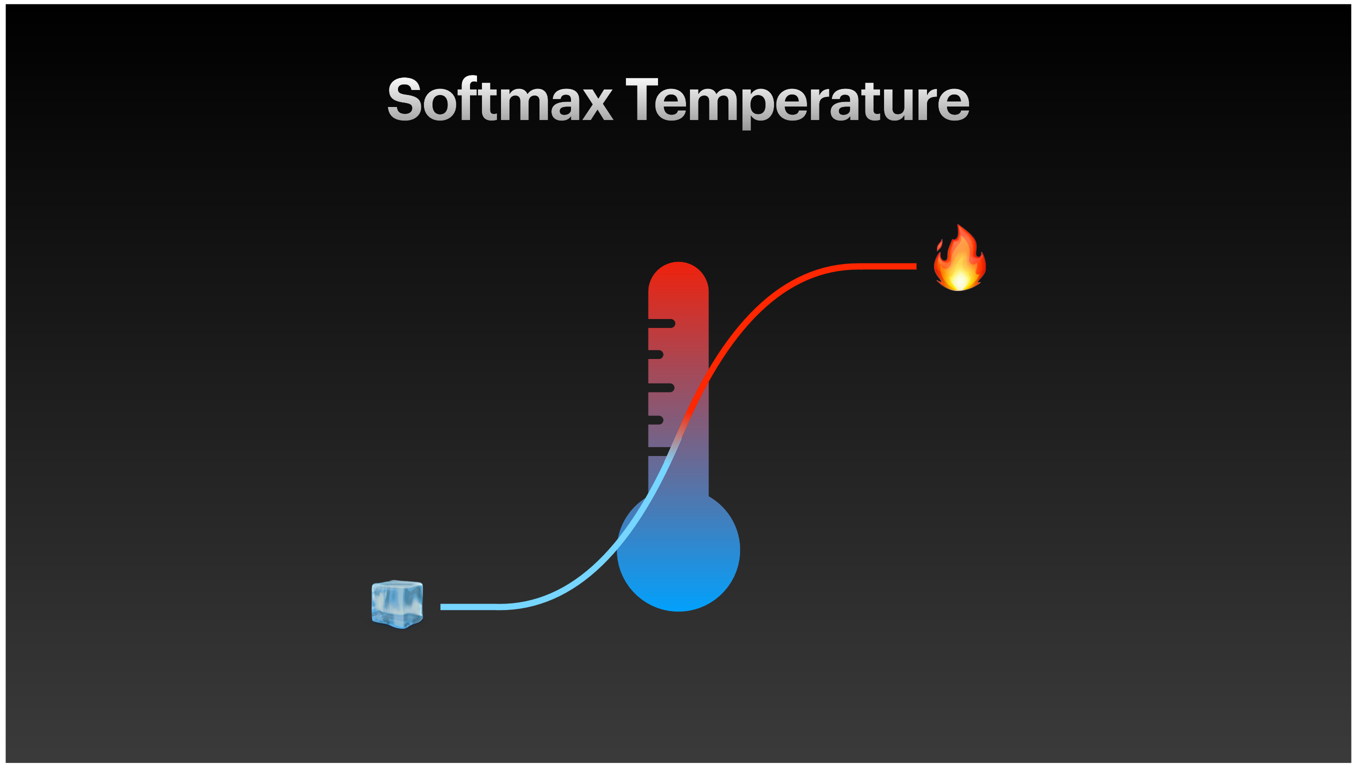

- The inference results are also known as “Logits.” The unnormalized predictions from a model
- Interpreting their values are difficult
- They are a value distribution, but not a probability distribution
- Values can be negative and/or much larger than one



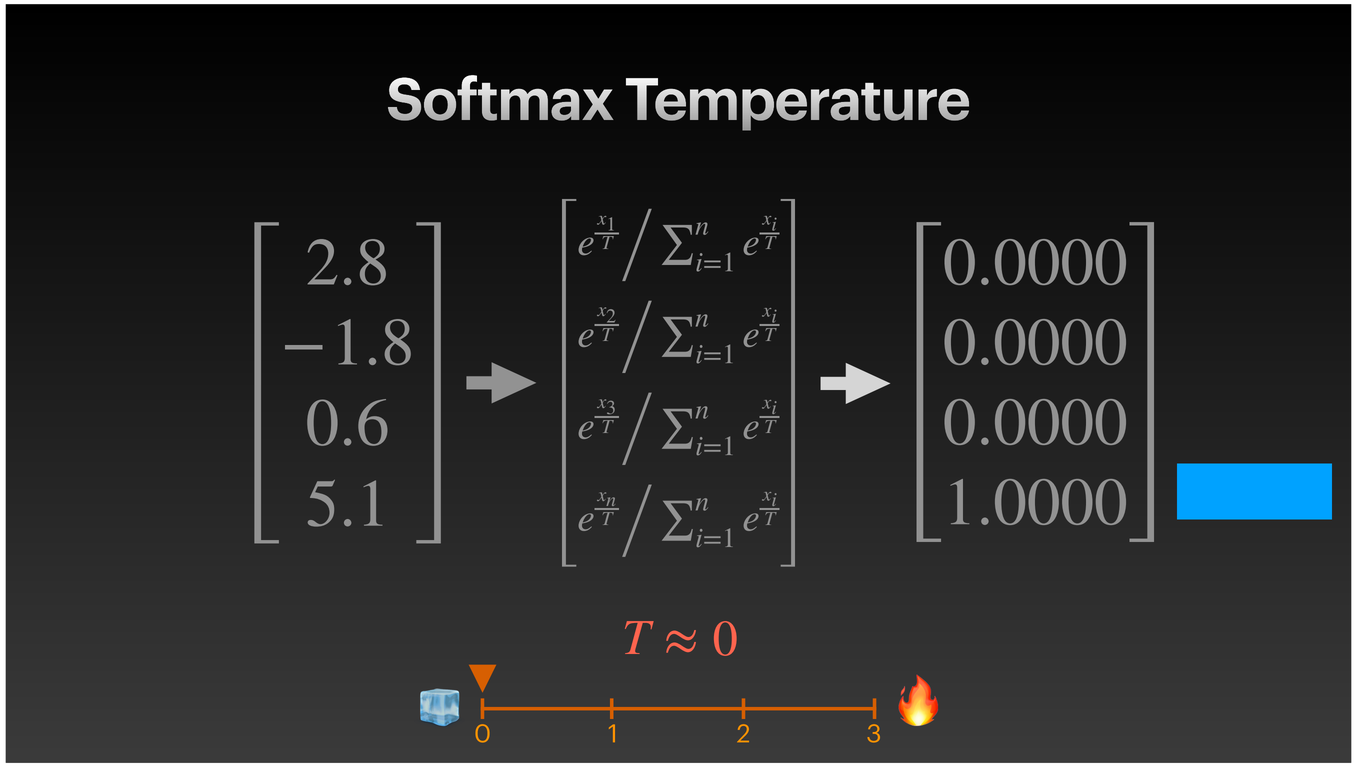

- Softmax is the standard way to turn a value distribution into a probability distribution
- The largest values are magnified and the smallest values are minimized




- Softmax normalizes the inference results into a probability distribution



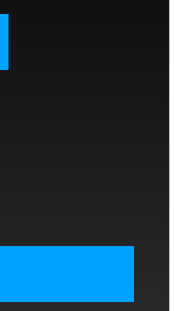

- The Softmax temperature varies the weights given to values from the inference outputs
- This opens the door for creativity, but also to hallucinations




- The temperature works by varying the denominator of the Softmax exponents be a value other than 1
- T < 1 will give more weight to the higher values (less creativity, fewer hallucinations). The probability distribution is concentrated on the maximum value
- T > 1 will give more weight to the lower values (more creativity, more hallucinations). The probability distribution is more uniform

# Softmax Temperature

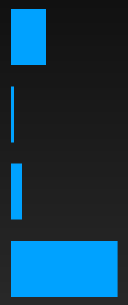
$$\begin{bmatrix} 2.8 \\ -1.8 \\ 0.6 \\ 5.1 \end{bmatrix} \rightarrow \begin{bmatrix} e^{\frac{x_1}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_2}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_3}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_n}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \end{bmatrix} \rightarrow \begin{bmatrix} 0.0901 \\ 0.0009 \\ 0.0100 \\ 0.8990 \end{bmatrix}$$



$T = 1$

# Softmax Temperature

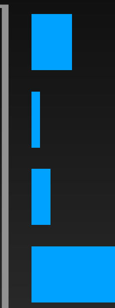
$$\begin{bmatrix} 2.8 \\ -1.8 \\ 0.6 \\ 5.1 \end{bmatrix} \rightarrow \begin{bmatrix} e^{\frac{x_1}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_2}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_3}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_n}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \end{bmatrix} \rightarrow \begin{bmatrix} 0.2178 \\ 0.0218 \\ 0.0725 \\ 0.6879 \end{bmatrix}$$



$T = 2$

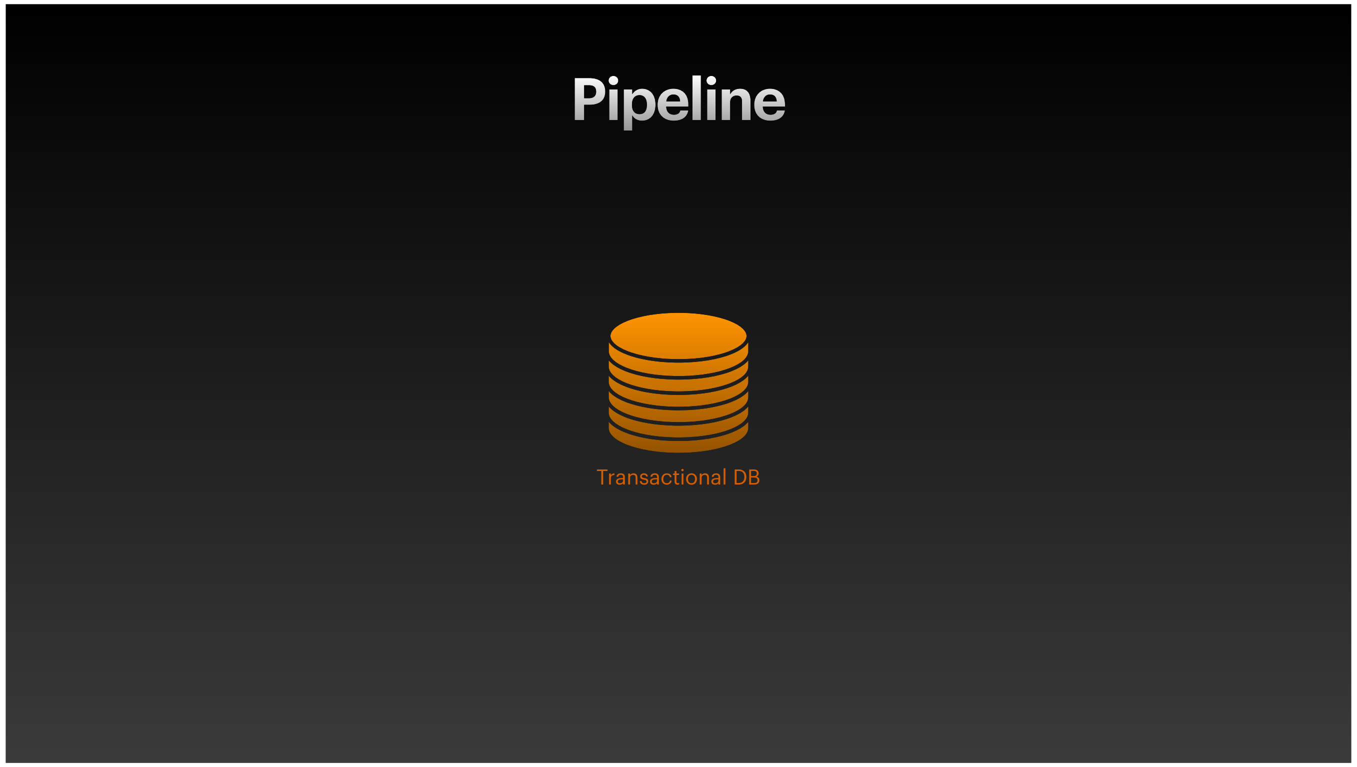
# Softmax Temperature

$$\begin{bmatrix} 2.8 \\ -1.8 \\ 0.6 \\ 5.1 \end{bmatrix} \rightarrow \begin{bmatrix} e^{\frac{x_1}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_2}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_3}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \\ e^{\frac{x_n}{T}} \Big/ \sum_{i=1}^{n} e^{\frac{x_i}{T}} \end{bmatrix} \rightarrow \begin{bmatrix} 0.2598 \\ 0.0561 \\ 0.1248 \\ 0.5593 \end{bmatrix}$$

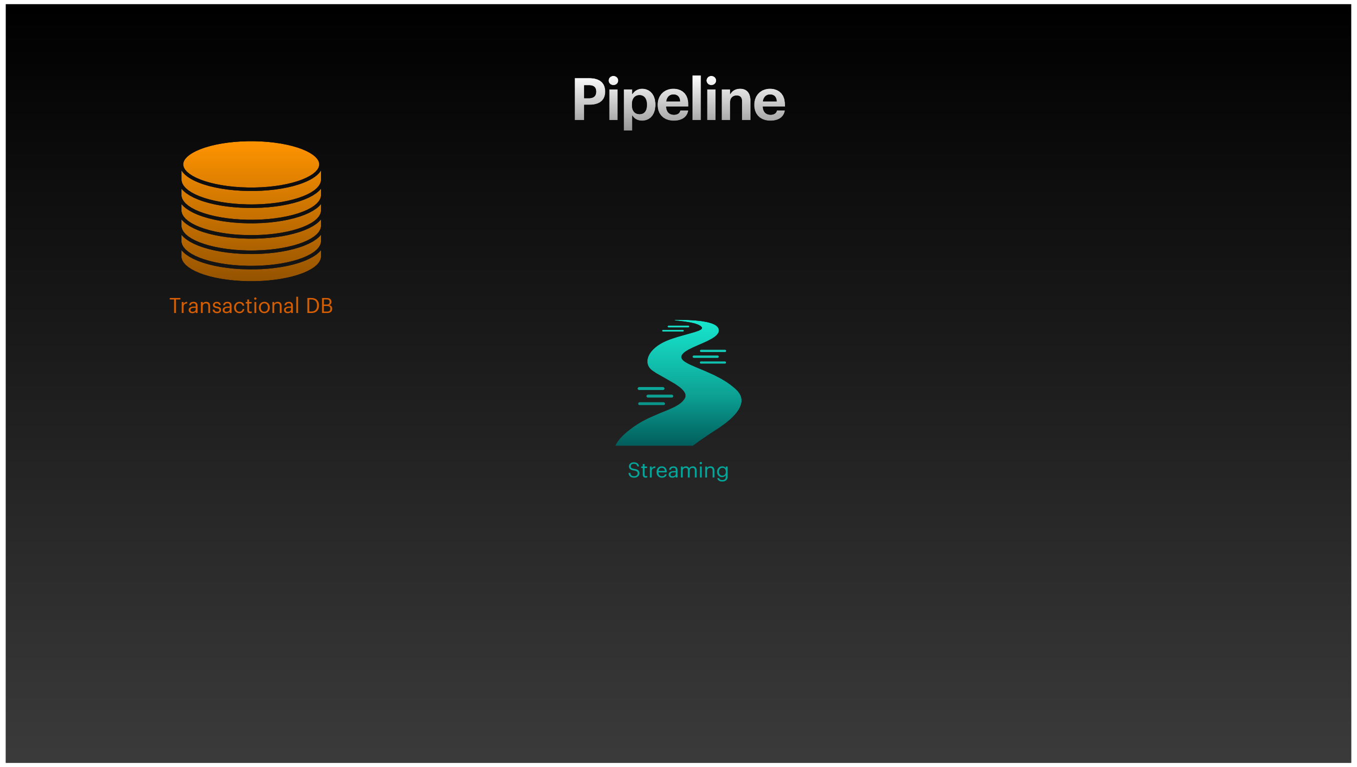


$$T = 3$$

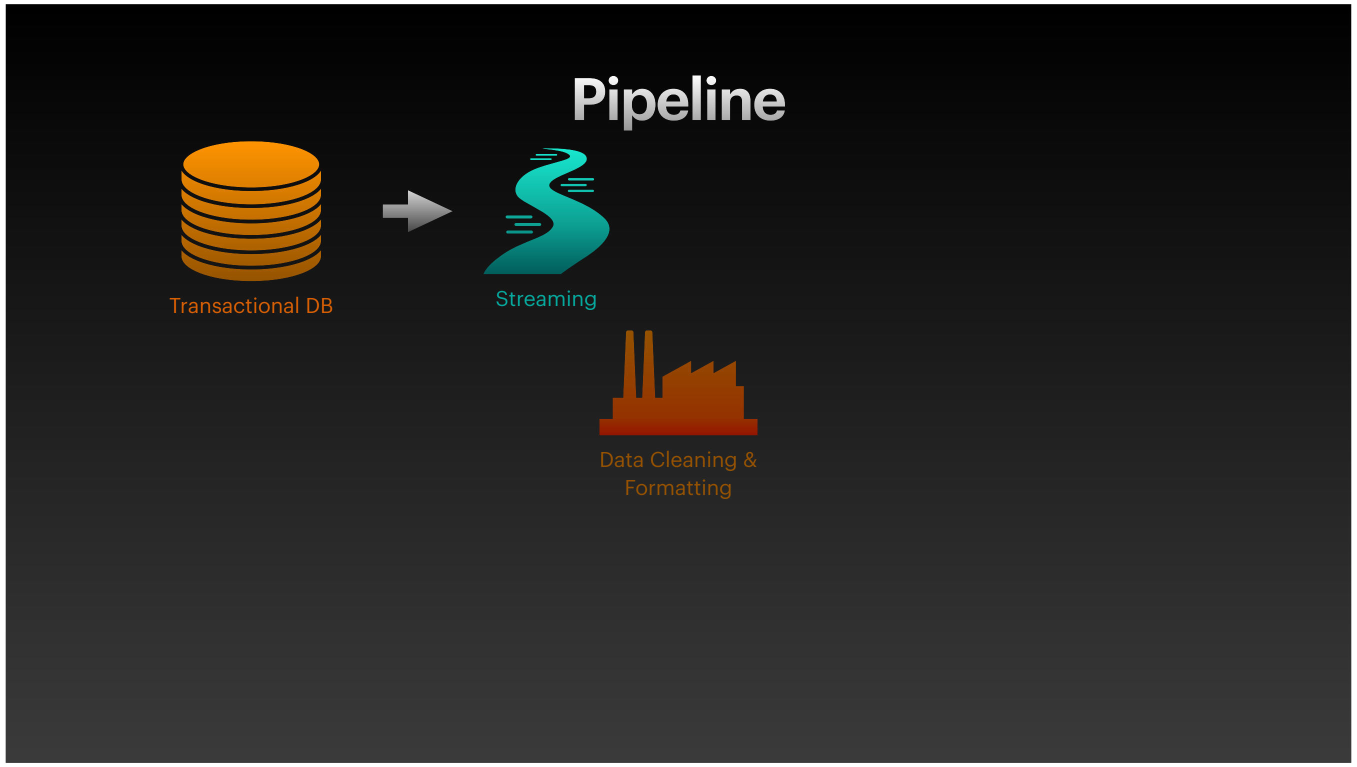

- In the demo we will cover how to do semantic chunking, how to generate embeddings from the chunks, store them in a vector database, retrieve the content, use it for RAG, prompt engineering, and how to put everything together to get better answers from an LLM
- Access the code at: https://github.com/dcirne/rag_fundamentals



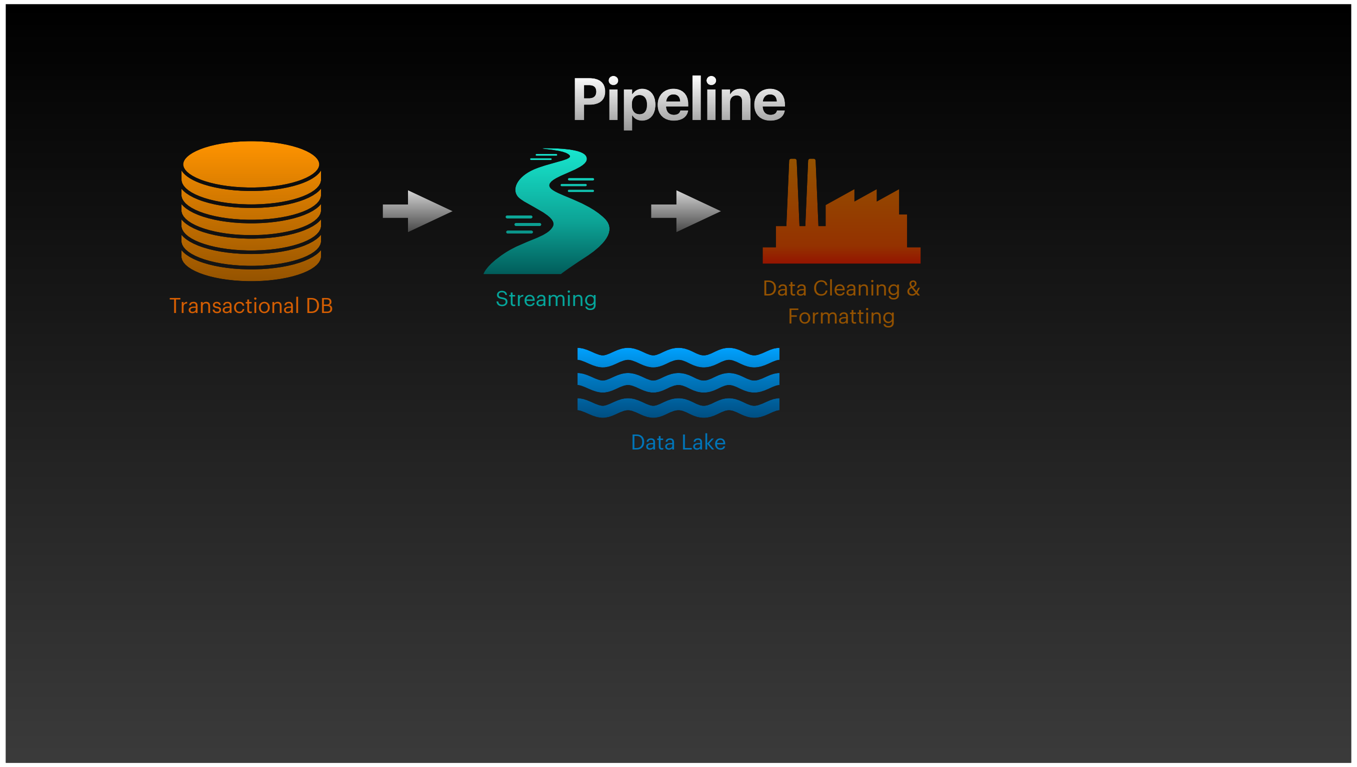

- Let’s switch gears and discuss how this can be deployed to large scale environments
- Although the transactional database may be the source of truth, operating ML directly on it may have severe performance consequences to users/customers



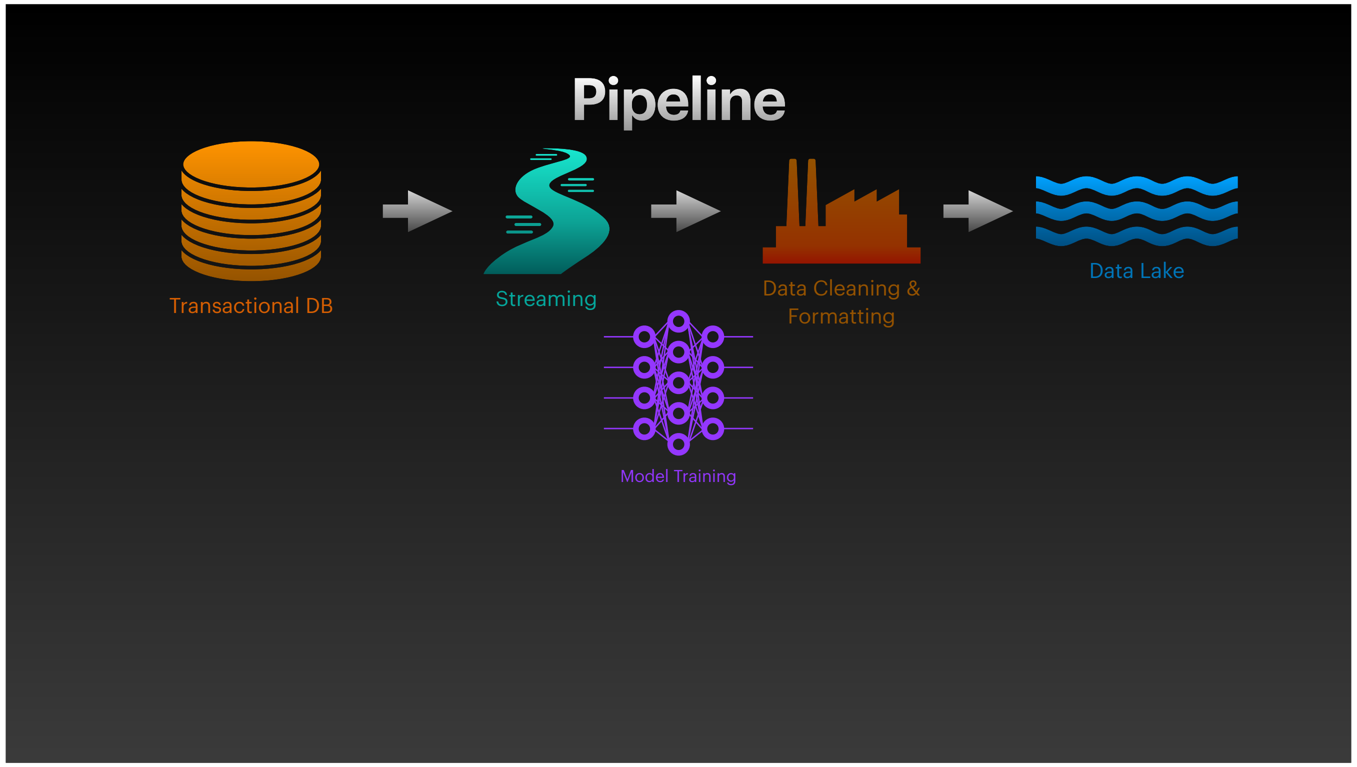

- The data needs to be extracted and replicated somewhere else
- Implementing an event streaming tool is the first step to extract data from the transactional database
- Changes in the data can be consumed in batches or send continuously as they happen
- Although an oversimplification, an event streaming tool is an implementation of the publisher/observer design pattern. That is perfect for the job at hand, since it will notify the observer when data needs to be processed



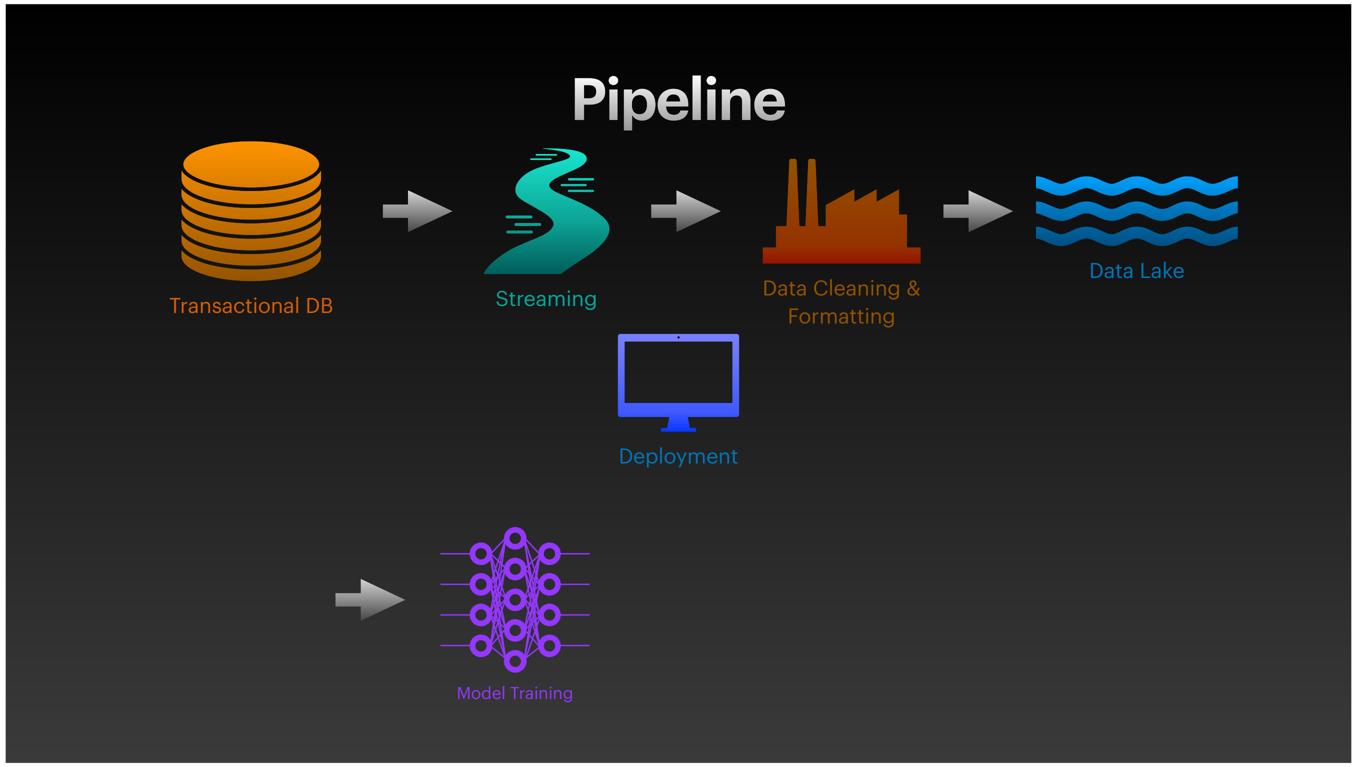

- But before the data can be used for ML, it may need to be cleaned (remove empty or incomplete rows) and formatted (from relational to tabular)
- Implementing a distributed computation tool is necessary for this step



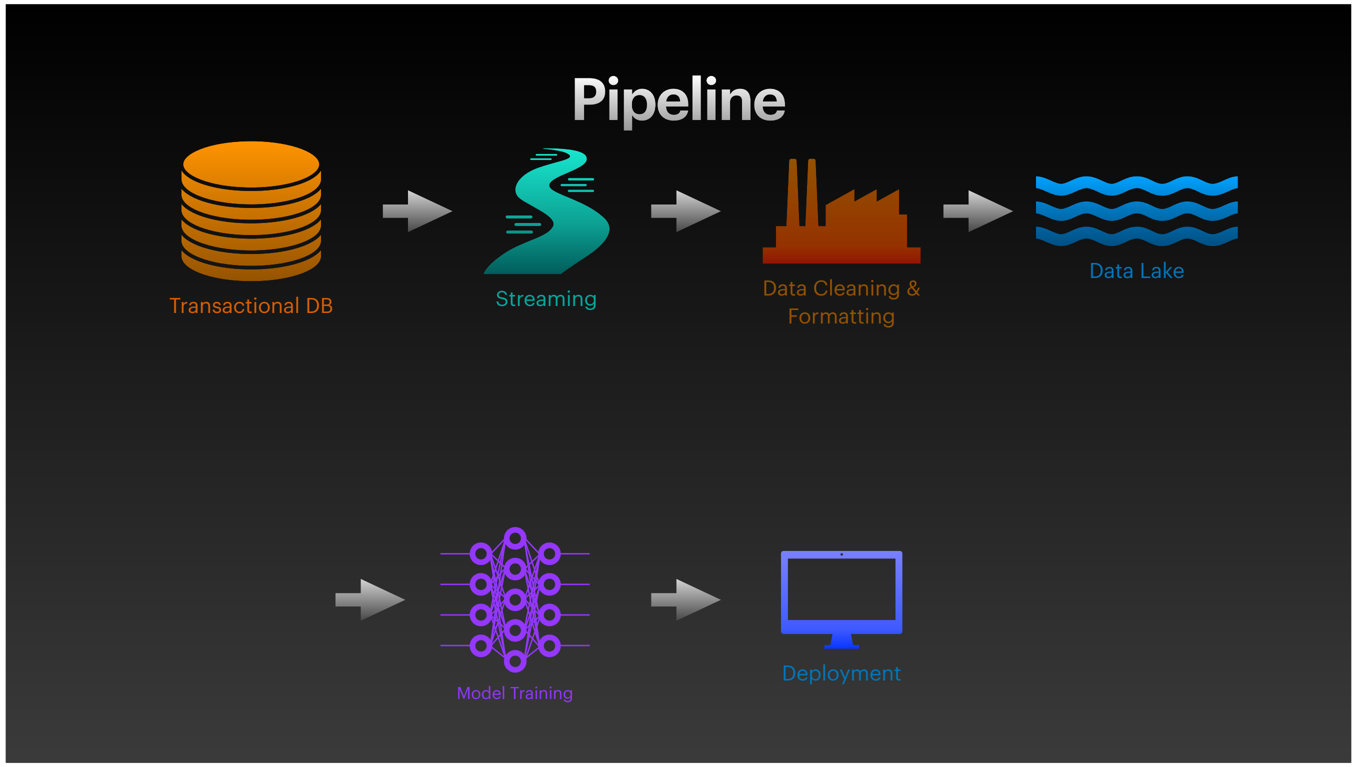

- The data is ready to be used for machine learning, but it needs to be stored somewhere
- A data lake tool becomes necessary, as they are optimized for these kinds of computations

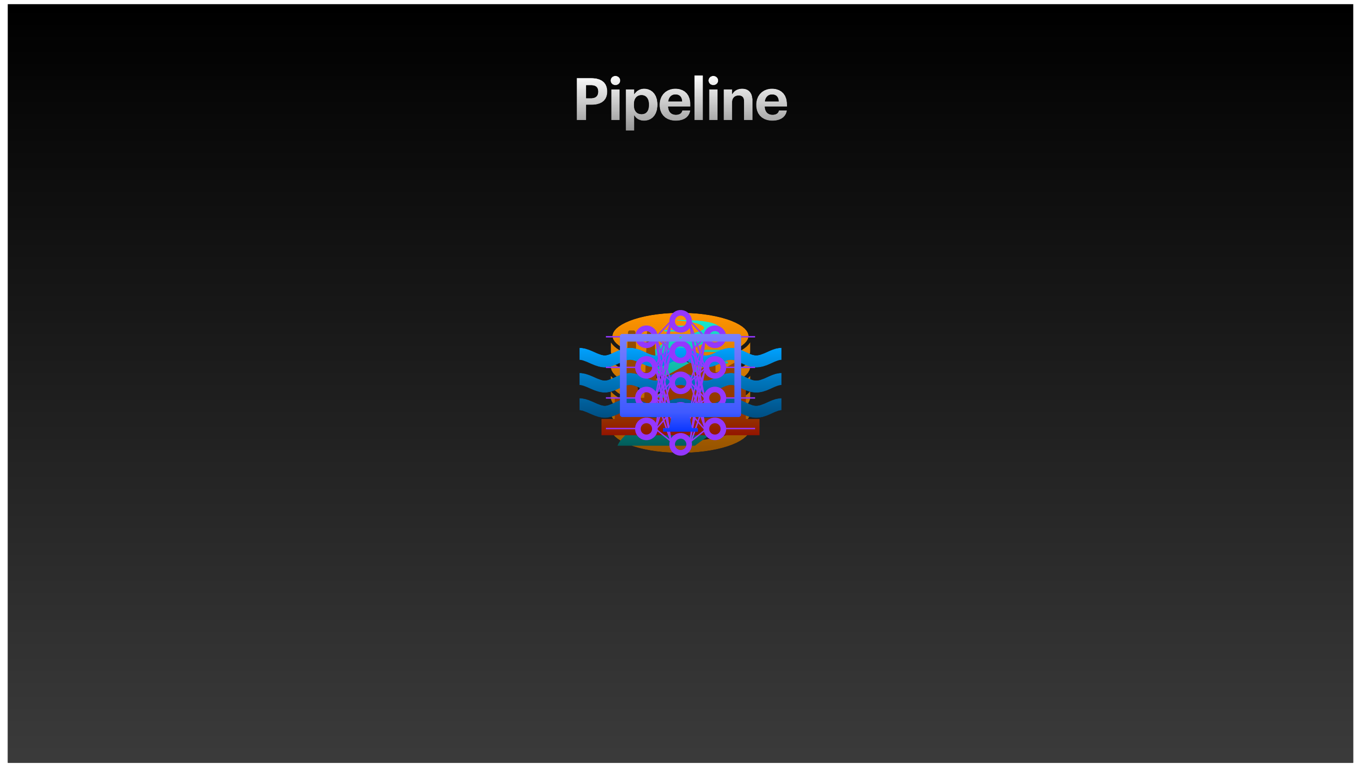



- Model training needs orchestrated automation
- Regular model retraining, multiple customers, and various data sovereign regions are unfeasible to maintain as a manual operation
- Manual model training should be reserved for research, special purposes, or an emergency situation

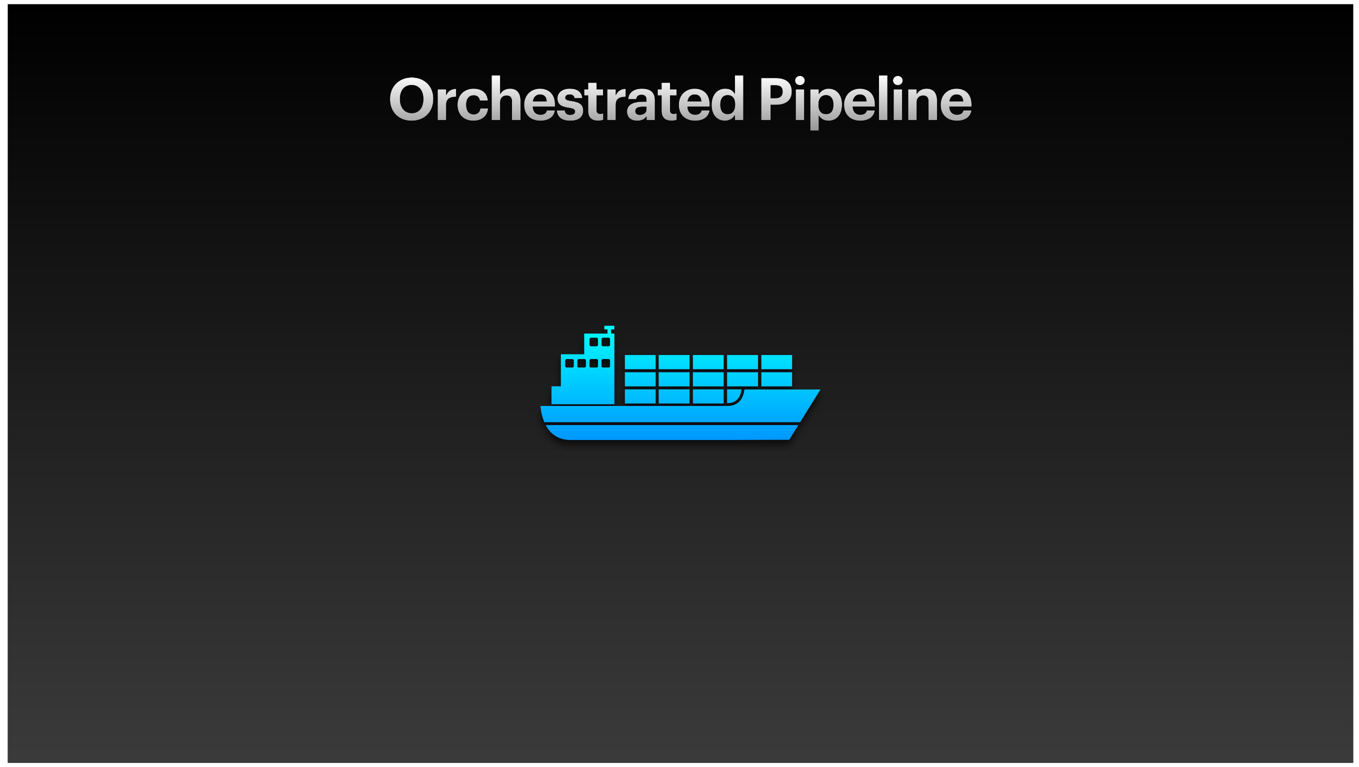



- Auto-scaling clusters, Remote Procedure Calls, Sharding, and other architectures need to be considered for deployment
- A single computer will not be able to handle all the traffic
- Replicating the same deployment won’t work, since models from all customers won’t fit in memory
- A combination of different options is more likely to be what is needed

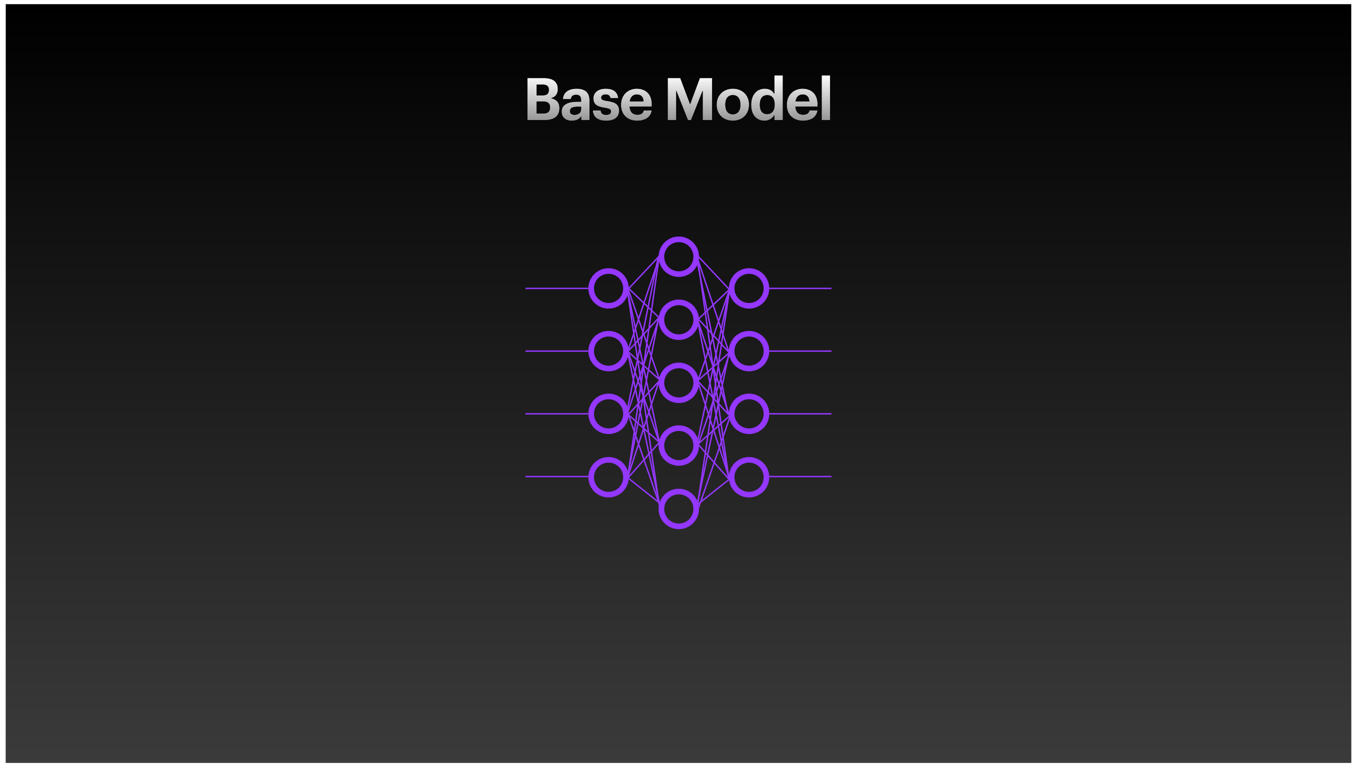

Pipeline
Transactional DB
Streaming
Data Cleaning &
Formatting
Data Lake
Model Training
Deployment



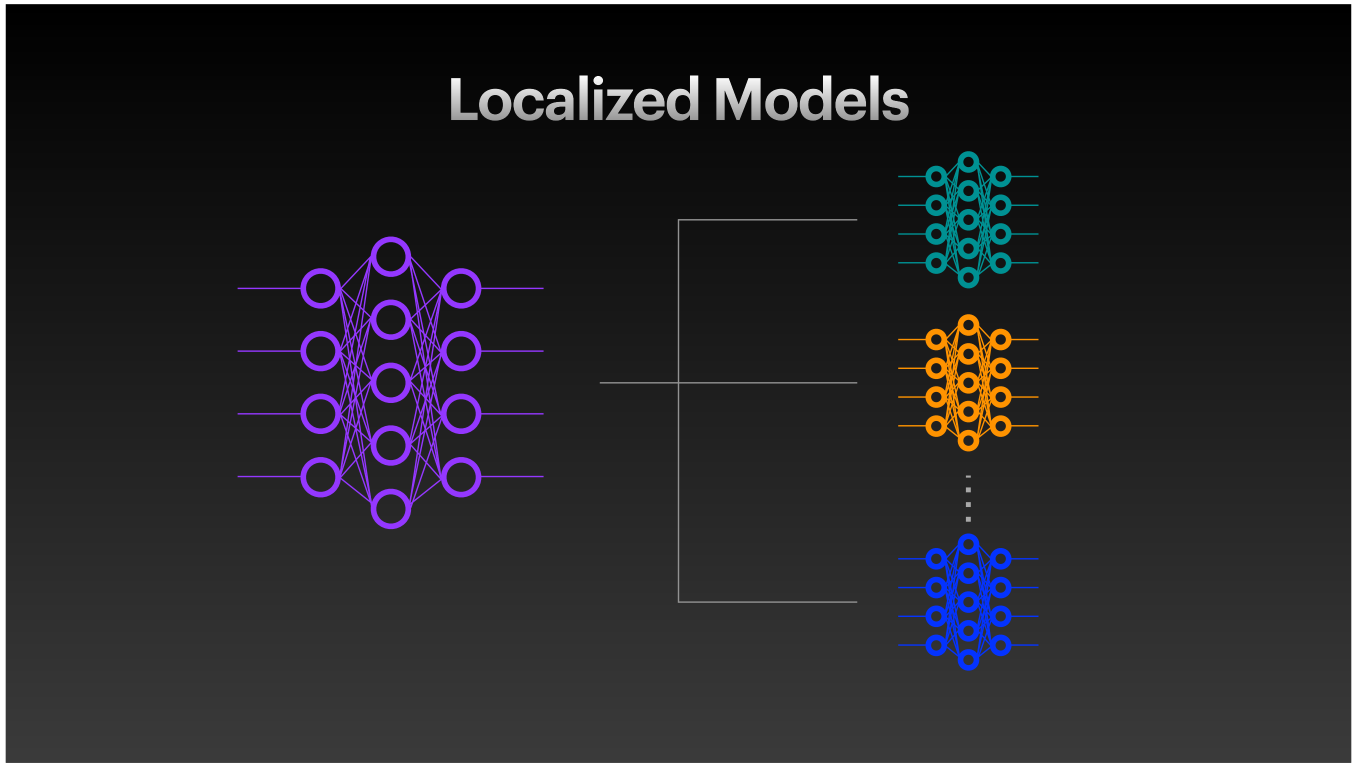

- The combination of all the elements together can be represented as an orchestrated pipeline



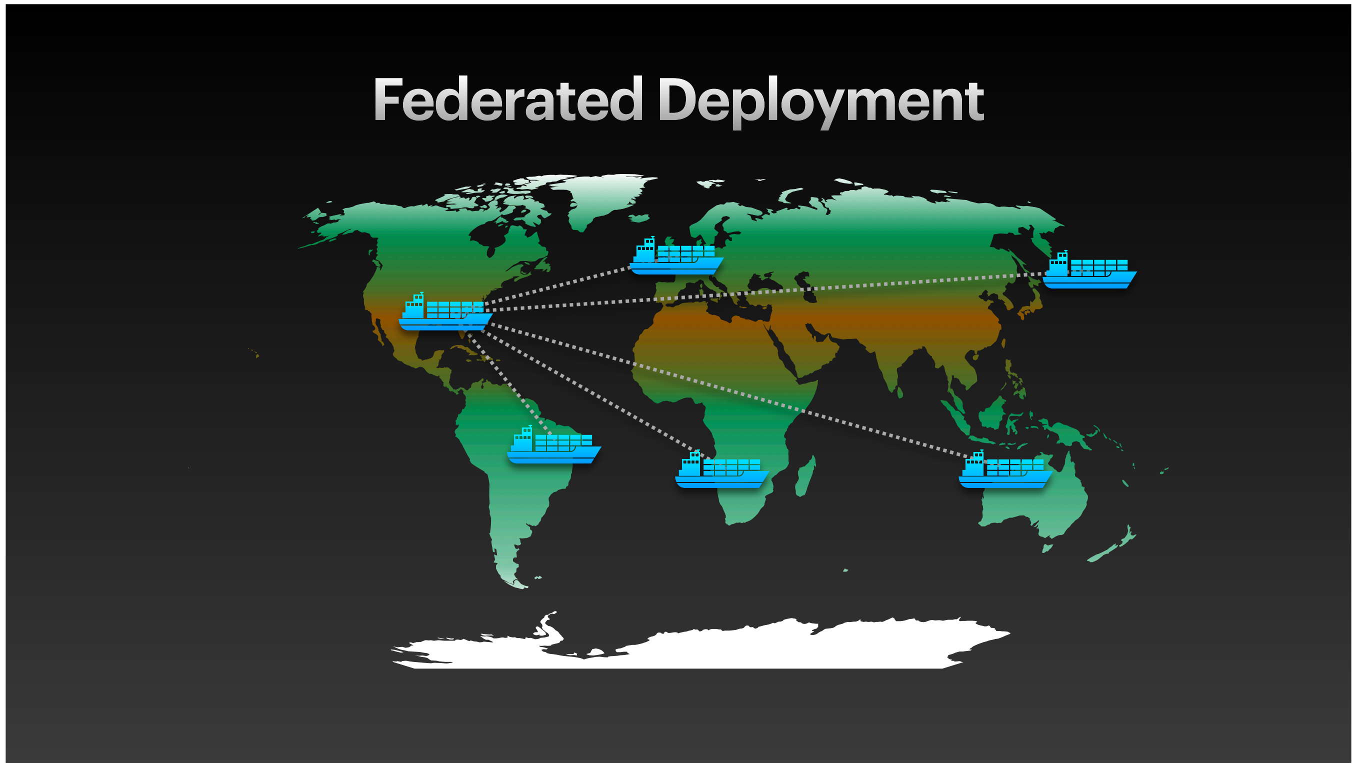

- Now this orchestrated can be replicated in a variety of situations, including, but not limited to disaster recovery locations, multiple regions, and places where data isolation is a requirement




- In many cases a base model can be trained with general purpose data that does not include customer content, personally identifiable information, or any other form of sensitive information
- For instance, concept of an expense report is a universal concept. It has a title, a date, line items, a total, and so on
- The same is true for a job posting. It has requirements such as skills, experience level, location, and so forth




- Base models can be trained with universal concepts
- Then later, other models can be trained on top of the base model using customer data or other restricted data
- At a minimum this saves time and energy, but it also allows for federated training




- Many regions/countries have data sovereignty laws or requirements
- The infrastructure has to be brought to those locations and it is not always possible to have personnel at those locales

# Let’s Talk. Here and Online.

Dalmo Cirne
https://dalmocirne.com

IARIA 2024